

Remarks

Claims 1-3, 8-16, and 21-26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application Publication US 2003/0005419 (Pieper) in view of *Portable Software Library Optimization*, 2/1998 (Cain).

Claim 1 recites at least the following limitations which claim 14 also similarly recites:

(a) optimizing the software program such that a *resulting first optimized form* of the software program is *completely independent of the target processor* and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program , and comparing the first performance profile with the performance objectives;

(emphasis added.)

The Board of Appeals and Interferences (the Board) has agreed with Applicants that Pieper does not disclose or suggest optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language. However, according to the Board, Cain allegedly discloses the above limitations.

A. No prima facie case of § 103 rejection

As an initial matter, Applicants respectfully note that a § 103 rejection cannot be sustained by merely alleging where the claim elements may be found in different references. Rather, the prima facie case of a § 103 rejection must include a motivation for combining the elements from the references in the purported manner. In this case, there is nothing in the record that indicates the motivation for combining the alleged element of Cain with the method of Pieper. For at least the foregoing reasons, Applicants submit that the prima facie case of the § 103 rejection has not been established, and request that the § 103 rejection be withdrawn.

B. Combination of references cannot be made in hind sight

In addition, Applicants note that it would be improper to use *hind sight* for combining references. In particular, it would be improper to use hind sight to find a reason to combine references after an element is found in a reference. In this case, regardless of whether Cain actually

discloses the above limitations or not, it would be improper to now retroactively use hind sight to look for a reason to combine the alleged feature of Cain with Pieper. Since the § 103 rejection that is based on the combination of Cain and Pieper cannot be established without using hind sight, Applicants submit that the § 103 rejection based on Cain and Pieper be withdrawn for this additional reason.

C. The § 103 rejection cannot be sustained if one reference teaches away from the feature of another reference.

Further, Applicants submit that since Pieper specifically describes an expanded code 56 in a form 60 that is “substantially independent of the architecture of the target processor 12,” Pieper actually teaches away from a resulting first optimized form of a software program that is completely independent of the target processor. In particular, figure 2 and paragraph 30 of Pieper specifically states that “front end process 54 translates source code 52 into a compact intermediate form 56. Code 56 is then processed by optimization processes 58” which “expands code 56 into an expanded intermediate form 60” which in turn “is substantially independent of the architecture of the target processor 12.” Furthermore, the product of the optimizer processes 58 – the expanded intermediate form 60 – is also not completely independent of the machine or process because, as paragraph 30 explicitly states, “[t]he code 60 output by the optimization processes 60 is in an intermediate level program code language that is substantially independent of the architecture of the target processor 12.” (Emphasis added.) Note that the Board has agreed with Applicants’ characterization of Pieper that “substantially independent” is still “dependent” and therefore, is not “completely independent.”

Appellants further respectfully point to paragraphs 17-18 in Pieper, where Pieper explicitly states its advantages over the prior art in that “[i]t is our contention that this consideration [of how far ahead to prefetch] should be made in terms of the cache memory itself (i.e., . . . to match the simultaneous request capability of the memory system)” and that “[t]he Alpha 21264TM processor dismisses prefetch instructions that hit in the on-chip cache Therefore, it is best that program code . . . be fitted with prefetch instructions” In addition, paragraph 19 of Pieper also explicitly states that “among the optimization process is a process that determines whether and where in the second set of instructions to insert memory prefetch instructions.” Based on these cited passages of Pieper, it is clear that not only is Pieper’s optimization process not completely independent of the target processor but it actually takes the processor and capability of the memory system into

consideration when optimizing the code. Since Pieper specifically describes how the optimized result is not completely independent of the target processor or the machine, Pieper actually teaches away from the above limitations, and therefore cannot be combined with Cain with the manner purported.

CONCLUSION

If the Examiner has any questions or comments regarding this response, please contact the undersigned at the number listed below.

The Commissioner is authorized to charge any fees due in connection with the filing of this document to Vista IP Law Group's Deposit Account No. 50-1105, referencing billing number **CA7017922001**. The Commissioner is authorized to credit any overpayment or to charge any underpayment to Vista IP Law Group's Deposit Account No. 50-1105, referencing billing number **CA7017922001**.

Respectfully submitted,

DATE: April 27, 2009

By: /Gerald Chan/
Gerald Chan
Registration No. 51,541

VISTA IP LAW GROUP, LLP
1885 Lundy Ave., Suite 108
San Jose, California 95131
Telephone: (408) 321-8663 (Ext. 203)
Facsimile: (408) 877-1662